IN THE CLAIMS:

Please amend claims 10, 14 and 15 and add new claims 16-28 as follows:

10. (Amended) The method for performing a group-multiply-and-sum instruction according to claim [8] 16, wherein the instruction comprises a [fixed-point] floating-point arithmetic operation.

14. (Amended) The method for performing a group-multiply-sum-and-add instruction according to claim [12] 17, wherein the instruction comprises a [fixed-point] floating-point arithmetic operation.

15. (Amended) A multiplier processing system for performing a group-convolve instruction, said system comprising:

means for partitioning each of a plurality of operands into a plurality of symbols, said operands having a first defined bit width and said symbols having a second defined bit width, said second defined bit width [of said symbols] being dynamically variable;

means for multiplying a selection of symbols of a first operand with a selection of symbols of a second operand, each of such multiplications producing a selected product, said selection determined by the indices of the symbols within the first and second operand as to perform a convolution; and

means for adding [each product] a plurality of selected products so as to produce a [single scalar result, said scalar result capable of being represented by a bit width which is equal to or less than said first defined bit width of said operands without a reduction in the accuracy of said

result] plurality of result symbols, said result symbols provided to a plurality of partitioned fields of a result operand.

16. (New) The method of claim 8, wherein said operands have a first bit width and said symbols have a second bit width, said second bit width being dynamically variable, and said scalar result is capable of being represented by a bit width which is equal to or less than said first defined bit width.

17. (New) The method of claim 12, wherein said operands have a first bit width and said symbols have a second bit width, said second bit width being dynamically variable, and said scalar result is capable of being represented by a bit width which is equal to or less than said first defined bit width.

18. (New) The multiplier processing system for performing a group-convolve instruction according to claim 15, wherein the instruction comprises a fixed-point arithmetic operation.

19. (New) The multiplier processing system for performing a group-convolve instruction according to claim 15, wherein the instruction comprises a floating-point arithmetic operation.

20. (New) The multiplier processing system for performing a group-convolve instruction according to claim 15, wherein the summation-tree of the multiplier array is utilized in a close approximation to the manner required for a scalar multiply.

21. (New) The multiplier processing system for performing a group-convolve instruction according to claim 15, wherein the multiplier array required for a scalar multiply comprises an accumulation array partitioned to form a plurality of sums of products.

22. (New) A method for performing a group-multiply instruction in a general purpose, multiple precision parallel operation programmable media processor, said method comprising:

partitioning first and second registers into a plurality of floating point operands, said floating point operands having a predefined bit width, wherein said predefined bit width is dynamically variable;

multiplying, in parallel, said plurality of floating point operands in said first register by said plurality of floating point operands in said second, each of such multiplications producing a product to provide a plurality of floating point products; and

providing said plurality of floating point products to a plurality of partitioned fields of a result ~~register~~.

23. (New) The method of claim 22 wherein each of said first and second registers are partitionable into four fields to hold four floating-point operands in parallel.

24. (New) The method of claim 22 wherein said first and second registers are 128 bit registers.

25. (New) A general purpose, multiple precision parallel operation programmable media processor for performing a group multiply instruction, said processor comprising:

first and second registers partitioned into a plurality of floating point operands, said floating point operands having a defined bit width and said defined bit width being dynamically variable;

a multiplier, configured to multiply, in parallel, said plurality of floating point operands in said first register by said plurality of floating point operands in said second register, each of such multiplications producing a product to provide a plurality of floating point products; and

a result register having a plurality of partitioned fields for receiving said plurality of floating point products.

26. (New) The processor of claim 25 wherein each of said first and second registers are partitionable into four fields to hold four floating-point operands in parallel.

27. (New) The processor of claim 25 wherein said first and second registers are 128 bit registers.

28. (New) The processor of claim 25 further comprising a multiplier configured to group multiply a plurality of fixed point operands in parallel.